Presentation by Dr. Eberhardt Rechtin, Internationally Renowned Systems Architect

Selective Interoperability
or
Making Interoperability a Practical Reality

Introduction

The National Airspace (NAS) System is one of the most difficult architectural tasks ever attempted by the United States Government for several reasons.

Its agency, the FAA, has very little control over its suppliers, its users, other government agencies and, most serious of all, of the airplanes it serves and the equipments they carry.

The system is, in reality, a "system of systems" by which is meant a group of semi-autonomous, self-standing, self-managing systems which, collectively are supposed to produce results which exceed the sum of their individual results. The largest of these semi-autonomous systems are aircraft avionics, communications, navigation, surveillance, and weather. The facilities of each are separately owned with different priorities and schedules. Yet, all are involved in air traffic control at and near airports, en route, and across oceans. They are coupled not only together but with international organizations and all three branches of the Federal Government; especially, the Department of Defense, NASA, NOAA, the Department of State, the Treasury, a variety of State and local agencies, the public, the media……and I've probably left out some comparably important ones.

This NAS "system of systems" is mandated to produce safe, efficient, cost-effective air travel in the interests of all the stakeholders. Note that <u>none</u> of the individual systems individually can do so. They must work together, in close to real time, to do so. The mechanism they use is information. The <u>structure</u> of information generation, processing, transfer, storage, display, understanding, use and response is called an information architecture. Its technologies are communications, computers, displays, satellites and software. Some of us understand this architecture by the term Command, Control, Communications, and Intelligence (C^3I). By whatever the name, it is the key to the <u>behavior</u> of any system of systems. It is the centerpiece of any and all <u>smart</u> systems of which the NAS is certainly one.

For many years, the DoD relegated C^3I to the status of a support system to the "important" things like bases, weapons, and platforms (airplanes, ships, and tanks). It was only in the 1960's that operations became so complex, distances so great, and timeliness so important that the crucial role of accurate, credible, and secure information was finally understood. C^3I is now <u>the</u> top priority of the DoD, hands down, no question, <u>top</u>. It is the crucial combatant in the Information War. I have yet to hear anyone argue with the fact that it won its part of the Gulf War overwhelmingly, greatly reducing the cost and casualties in the other systems. It wasn't perfect, but it was convincing. Yet it was flexible enough that the commanders on the spot could completely reconstruct it *in the field* when confronted with one of the most dangerous air traffic control problems yet - keeping armed aircraft from many nations, including an attacked friend, Israel, all too anxious to retaliate - from annihilating each other by mistake. Complicating the ATC problem were subsonic stealth aircraft that were supposed to fly right through this air space without being shot down by others within visible range!

So, don't tell me that an Information System is just supposed to put out the daily weather report and to make sure that the right male electrical and communication plugs fit into the right female sockets, so to speak.

But, at the same time information systems became joint operators and combatants, they became life-threatening, vulnerable, destructive and error-prone. In procurement, particularly of software, they have blown costs and schedules. In behavior, they have created mathematical chaos. In smart systems, they have become too smart for their own good. These systems have all too often provided examples of the Law of Unintended Consequences. After all, they are designed and operated by 1% error-rate human beings.

The technical reason that information systems create such difficulties is because the individual elements must communicate with each other about very important, often urgent operational matters - and this communication must be accomplished *easily, securely, accurately, and certifiably, all on demand.* If, for any reason, the agencies can't or won't talk, if the messages can't be understood or take too long to arrive, or if somebody who should, but doesn't, get the word, all hell breaks loose. If the information systems interfere with the essential tasks of the elements, they will be resisted, to the damage of all parties.

I. The difficulties with everything everywhere, all-the- time, for everybody.

The first thought in resolving such problems is to have all the systems open to each other through the use of common equipments and procedures. Theoreticians in the procurement community love this idea. Unfortunately, it isn't practical for near-autonomous, separately owned, systems doing essential functions for others at their own time and pace; in effect, there is no single date at which all "old" systems can be taken out and "new ones" installed without very high cost and considerable disruption. Worse yet, it is rare that any one system can be completely changed without affecting the systems to which it is connected. I know. We tried it for military satellite systems. We tried it with the NATO nations.

A simple example. Suppose the computers in use by system A are all Macintoshes and all those in use by system B are IBM clones and someone in Information Systems decides that the new common system should use NExT computer. (Don't laugh. I faced one like that in WWMCS when it was decided to use Honeywell - which subsequently went out of business.) What then do A- and B do about their peripherals? Well, better change them, too! And that affects the surveillance sensors, the integration with the LAN's and Internets.... and so it goes. Information Systems, in particular, simply cannot mandate that common equipments be bought and used by the other, semi-autonomous systems. As will be seen later, it isn't necessary, anyway.

More important, commonality runs into security, proprietary, and specialization obstacles. In security-sensitive situations, too many people have access to too much information. In proprietary ones, too many competitors will use the system for what is euphemistically called "business intelligence." In areas of specialization, the education necessary to use much of the information in each technically different system is so extensive that few others can begin to understand it. But the most important obstacle of all - and it is highly subjective - is that of privacy. In a completely open environment, everyone can know anything desired without the knowledge or control of the "owner." That is, "Everything you know, I can know. All the information you own is mine." In a system in which the Congress, the media, other agencies, and your enemies have full access.....yeah, sure, even if you have the time, resources and the patience to answer every damn fool inquiry. NO!

So, what happens. Each system, in self-defense, procures equipments that can’t talk to others, like Army, Navy, and Air Force avionics. I can remember when that was tried, and lost in the early 1970's. It was so discouraging I haven't revisited that imbroglio since.

II. The real needs when one gets down to it.

Miserable. Tragic. Because, in the end, unless these "systems within a system" work together, all will fail together. They must communicate. BUT, as I said earlier, "easily, securely, accurately, and certifiably, all on demand. "

I didn't say, "using common equipment and data bases, by specific networks, on any subject, and all the time." Those are not needs, they are some proponent's idea of how to satisfy them. There is an enormous difference between say, "all the time" and "on demand," in what the architectural construct must be in order to satisfy them. For examples, consider the relative ease of providing on-demand security, and/or on- demand certification.

"Full interoperability" demands are not only unnecessary, they are undesireable and destructive technically and psychologically. For example, any design that provides full access also produces saturation of the reader. I can remember the effect of the FleetSatCom on the Navy ships at sea. That satellite system provided UHF, 25 hertz channels to replace high frequency radio telegraphy and voice. It was a modest capacity, but those well-meaning people ashore sent so many messages that it took all the officers 24 hours per day simply to read them! The system was saturated at its end terminals, with the resultant missed messages, late replies and confusion.

So what did the recipients do? They discarded all messages, without reading them, except for those from recognized sources addressed directly to them, individually. They soon resented long messages because they often were addressed to others and delayed those addressed to them. So it was proposed, though I don't know whether it was implemented, that no message could be more than a certain length and anything longer had to go back to the end of the que. Another example of more immediate importance to the NAS, Smith et al in an IEEE article'[1] showed that there were powerful effects on pilot's flight plan depending on <u>how</u> the computer systems supported them.

Of course, technologists could fix all that. Simply increase the bandwidth, put in more data circuits, provide data dumps at high speed - but often at the cost of voice communications, of course... the only communication that really provides automatic credibility of the sender and clear (voice) acknowledgement by the receiver.

III. *Scope. Scope. Scope* and other related heuristics.

These real needs can be satisfied not by "commonality" or "interoperability" but by what might be called "selective interoperability" To the systems architect, this suggests:

Scope! Scope! Scope!

In other words, first try to reduce the size of the problem by deciding, with the other stakeholders, just which, and to what degree, needs can be met in a practical manner, <u>and which can not</u>. Do all parties understand and agree with the essential needs of the other stakeholders and why? Are all agreed on who will be responsible for maintaining which data base, replying to queries to it, and how that reply will be

[1] Smith, et al, "Brittleness in the design of cooperative problem solving systems", IEEF, Transactions on Systems, Man and Cybernetics, Vol 27 # 3 May 1997.

presented to each inquirer? It is astonishing how much these agreements can reduce the "requirements" list. Then, the next step:

Simplify. Simplify. Simplify.

Is the information in its simplest, most direct, most easily understood form? Is there a simple way of inquiring further in case more is desired? Are meanings of words standardized and is there a glossary that states those meanings? Are there simple rules about message lengths to avoid link saturation or to provide system resiliency? Is there a much simpler, but perhaps less capable, system that can provide almost as much information sooner, or instead? Can any machine process the message in a locally-acceptable format (ASCII comes to mind)?

When the needs seem well scoped and about as simple as possible, then:

Group elements that are strongly related to each other.
Separate elements that are unrelated.
Choose a configuration that needs minimal communications between subsystems.
Never aggregate systenu that have a conflict of interest.

At this point there should be a well aggregated, well partitioned architecture designed to work. However, such architectures can be brittle; that is, they are not designed to fail (properly). Failing improperly means failing catastrophically, failing in a way that can not be diagnosed promptly, failing when most needed, etc. For this contingency,

Provide dissimilar redundancy for all critical functions.
(The Navy Ship Design equivalent: *All spaces will have two exits.)*

This heuristic means that whenever possible, be able to perform all critical functions in at least two different ways; e.g., by data or voice, by GPS or radar, by satellite or microwave relay, by different protocols, by alternate weather sources, by different airports, by different computer programs - all selectable through an Information System that responds at the touch of a button. In any ease, as is characteristic of all airliner designs,

There must never be a single point failure of the NAS as a whole,
its information system included!:

IV. <u>Thoughts on applicability to the NAS and its legacy systems, in particular</u>.

If you don't understand the existing system,
you can't be sure you are architecting a better one.

Unless constrained, rearchitecting has a natural tendency to proceed unchecked
until it results in a substantial transformation of the system.

Given a change, if the anticipated actions don't occur,
then there is probably an invisible barrier to be identified and overcome.

Don't try to do everything, much less all at once!
It isn't needed!

Presentation by Dr. Eberhardt Rechtin, Internationally Renowned Systems Architect (continued)

Thank you, even if I have messed up part of your agenda! I leave you with a possibly disquieting thought. The NAS is one of the easier government_systems to privatize, should that be perceived as worthwhile or, heaven forbid, necessary. So let's do it right.

Presentation by Dr. Eberhardt Rechtin, Internationally Renowned Systems Architect (continued)

Scope! Scope! Scope!

Group elements that are
strongly related to each other.

Separate elements that are unrelated.

Choose a configuration needing
minimal communications between its
subsystems.

Never aggregate systems that have a
conflict of interest.

Presentation by Dr. Eberhardt Rechtin, Internationally Renowned Systems Architect (continued)

Simplify. Simplify. Simplify.

Provide dissimilar redundancy
For all critical functions.

(The Navy Ship Design equivalent:
All spaces will have two exits.)

There must never be
a single point failure of the NAS as a whole,
its Information System included!

If you don’t understand the existing system, you can’t be sure you are architecting a better one.

Unless constrained, rearchitecting has a natural tendency to proceed unchecked until it results in a substantial transformation of the system.

Given a change, if the anticipated actions
don’t occur, then there is probably
an invisible barrier to be identified
and overcome.

Don't try to do everything,

much less all at once!

It isn't needed!

References:

Rechtin, E., *Systems Architecting, Creating & Building Complex Systems.* Englewood, NJ. Prentice Hall, 1991

Rechtin, E., and Mark W. Maier, *The Art of Systems Architecting*, Boca Raton, FL. CRC Press, Inc., 1997